Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

CRT-2003-28

ON SOLVING THE TREE-TOPOLOGY DESIGN PROBLEM FOR WIRELESS CELLULAR NETWORKS

by

Yanick Pomerleau
Steven Chamberland
Gilles Pesant

October 2003

Université de Montréal
C.P. 6128, succursale Centre-ville, Montréal QC H3C 3J7 Canada
Téléphone: (514) 343-7575 / Télécopieur: (514) 343-7121

crt@crt.umontreal.ca
www.crt.umontreal.ca

HEC ÉCOLE POLYTECHNIQUE MONTRÉAL Université de Montréal

104-00344/

PUBLICATION
CRT-2003-28

ON SOLVING THE TREE-TOPOLOGY DESIGN PROBLEM FOR WIRELESS CELLULAR NETWORKS

by

Yanick Pomerleau [1]
Steven Chamberland [1,2]
Gilles Pesant [1,2]

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

1. Computer Engineering Department, École Polytechnique de Montréal, C.P. 6079, succursale Centre-ville, Montréal, Canada H3C 3A7 (yanick.pomerleau, steven.chamberland, gilles.pesant@polymtl.ca)

2. Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2003

Abstract

In this paper, we propose a constraint programming model for the design problem of wireless cellular networks. It consists of selecting the location of the base station controllers (BSCs) and mobile service switching centres (MSCs), selecting their types, designing the network tree-topology and selecting the link types, while considering the location and the demand of base transceiver stations (BTSs). Next, we develop a heuristic combining local search and constraint programming techniques to find a good solutions in a reasonable amount of time. Numerical results show that our approach, on average, improves the results from the literature.

Keywords: cellular wireless networks; topological design; constraint programming; local search.

Résumé

Dans cet article, nous proposons un modèle de programmation par contraintes pour le problème de conception du sous-système réseau d'un réseau cellulaire. Ce problème consiste à déterminer la localisation et les modèles des BSCs (Base Station Controllers) et des MSCs (Mobile Switching Centres) à installer, la conception topologique du sous-système réseau et les modèles des liens à installer. Une heuristique combinant la recherche locale et des techniques de programmation par contraintes est proposée. Les résultats montrent qu'en moyenne cette heuristique est meilleure que les autres proposées dans la littérature.

Mots-clés: réseaux cellulaires; conception topologique; programmation par contraintes; recherche locale.

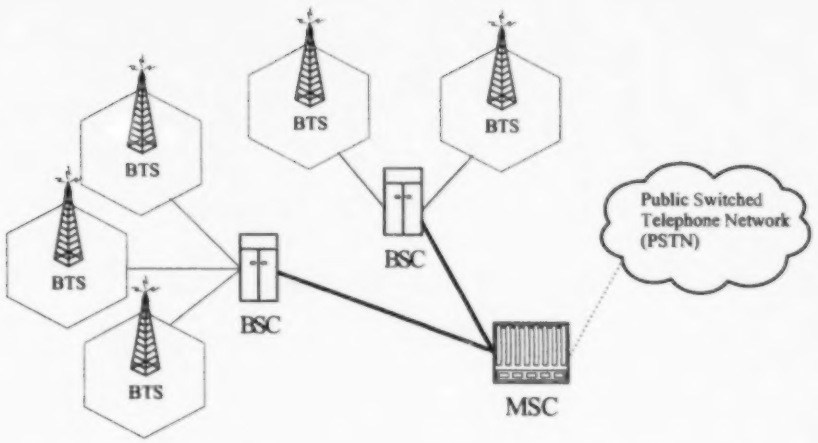

Figure 1: The network sub-system of a cellular wireless network

1 Introduction

In a typical wireless cellular network, the area of coverage is geographically divided into cells and the network topology is hierarchically organized in order to reduce costs. Each cell is equipped with a base transceiver station (BTS) that contains the radio transceivers providing the radio interface with mobile stations. One or more BTSs are connected to a base station controller (BSC) that provides a number of functions related to resource and mobility management as well as operation and maintenance for the overall radio network. One or more BSCs are connected to a mobile switching center (MSC) or switch that controls call setup and call routing while performing many other functions provided by a conventional communications switch. An MSC can be connected to other MSCs or networks such as the public switched telephone network (PSTN), in order to provide a larger coverage.

Cellular wireless network service providers dedicate an important proportion of their budget to acquire, install and maintain the facilities that carry traffic from cell sites to switches and other facilities (Figure 1). Typically, the design of cellular wireless networks requires:

1. the analysis of radio-wave propagation and/or the field topology to identify a set of possible base stations locations;

2. the selection of a least-cost subset of locations (network nodes) as hubs where the traffic is to be aggregated and switched;

3. the assignment of each cell to a switch while taking into account a certain number of constraints including capacity constraints, routing-diversity to assure reliability, handoffs frequency, and so on;

4. the selection of the type of links between the nodes or network elements.

This paper focuses on points 2, 3 and 4, we are interested in the global design problem of such networks. The proposed model deals with: selecting the location of the BSCs and MSCs; selecting the BSC and MSC types; designing the network topology; selecting the link types.

We make the following assumptions about the organization of the network: (C1) assignment constraints specify that each component is connected to exactly one other, such as a BTS to a BSC and a BSC to a MSC; (C2) uniqueness constraints specify that at most one component is installed at each site; (C3) capacity constraints are applied on links, BSCs and MSCs; (C4) traffic flow conservation constraints should be satisfied.

The following information is supposed to be known: the location of the BTSs and their types; the traffic (in erlang) between the BTSs togetter and between the BTSs and the public network; the location of potential sites to install the BSCs and the MSCs; the cost and the capacity of the components (BSCs, MSCs and links) of different types.

The objective is to minimize the cost of the design, including the cost of the equipment and their installation.

Many aspects of the overall design problem correspond to well-known operational research problems, such as graph partitioning [13, 14] or p-fixed hubs location [18]. Since these problems are NP-hard, exact algorithms are inappropriate in practice for moderate and large size instances. As a result, heuristic approaches have been largely used for solving these aspects of the design problem [2, 3, 5, 6, 13, 14, 15, 16, 18].

The literature abounds on network planning and the optimization of cellular wireless networks [4] but few have adressed the design problem as a whole. Merchant and Sengupta [13, 14] studied

only the assignment problem. Their algorithm starts from an initial solution, which they attempt to improve through a series of greedy moves, while avoiding to be trapped in a local minimum. The moves used to escape a local minimum explore only a very limited set of options. These moves depend on the initial solution and do not necessarily lead to a good final solution. Other heuristics, strongly inspired by that of Merchant and Sengupta, were proposed by Saha, Mukherjee and Bhattacharya [17]. These heuristics are based on the formation of cell clusters related to the same switch. The cell where the switch resides is the root of the cluster, then each cluster is extended by judiciously adding other cells. Several versions of the algorithm were proposed. In general, these algorithms improve the results of Merchant and Sengupta but nevertheless remain ineffective for designing large size cellular wireless networks. Andr, Pesant and Pierre [2] also studied the assignment problem and used a variable neighborhood search metaheuristic [8] to solve it. The neighborhoods are based on reassignment and redistribution of the BTSs to the switches. The initial solution is found by constraint programming. This algorithm improved the solutions found by the other heuristic methods such as simulated annealing [16] and tabu search [15] in quality and in CPU time. Sohn and Parc [18] studied the p hub location problem that consists of finding the location of p hubs and the cells assignment so that the total transportation cost is minimized. The authors have proposed a mixed integer formulation to solve the problem. The computational results show that the model works well. Only few instances provided non-integer solutions, less than 0.02%. For those instances, integer solutions are obtained quicktly with a branch and bound method.

Cox and Sanchez [6] studied the whole design process and used a tabu search metaheuristic, with embedded knapsack and network flow subproblem to design a least-cost telecommunications network to carry cell site traffic to wireless switches while meeting survivability, capacity, and technical compatibility constraints. In this context, each optimization problem is solved separately while taking into account its impact on the other ones. Because, they added some network survivability constraints, some switches must be assigned to at least two others switches. Their results illustrated that optimal solutions can be found. The cost difference between 20% and 100% of switches with survivability constraints is very small : less than 1%. The global design problem

defined here was considered by Chamberland and Pierre [5]. Moreover, we used the same set of instances to evaluate our approach. The authors proposed a mathematical programming model for this problem, demonstrated that it is NP-hard and proposed a tabu search algorithm. The tabu algorithm determines a set of BSCs and a set of MSCs to move from a solution to another while taking into account tabu moves and the aspiration criterion. The results showed that good solutions can be found with this approach (within 5.79%, on average, from a proposed lowed bound) in a reasonable amount of CPU time (1053 seconds, on average).

The lower bound used to evaluate the quality of the solutions obtained by this TS algorithm comes from an exact algorithm applied to a relaxation of the problem. Therefore it is difficult to know what proportion of the gap is an optimality gap. In this context, it is interesting to investigate other avenues to confirm and improve the quality of the solution obtained by the TS algorithm. The tree-topology design problem is composed of constraints that are easily expressed with finite domain variables. Hence, constraint programming (CP) [12] is an interesting option for solving the problem.

In this paper, we explore such an avenue based on constraint programming to solve the design problem of wireless cellular networks. The paper is organized as follows. Section 2 presents the constraint programming model for the global design problem of cellular wireless networks and, in Section 3, a heuristic is proposed. Section 4 presents numerical results and we conclude in Section 5.

2 The Constraint Programming Model

The application of CP to the resolution of combinatorial problems rests on the notion of finite domain variables. For each such variable, the (finite) set of all possible values it may take, called its domain, is saved and updated as the computation progresses. Constraints are imposed on the combinations of values these variables may take and are enforced using specialized filtering algorithms, either built-in or user defined, that work by removing inconsistent values from the domains. Ideally, the algorithm should remove all inconsistent values, that is, those for which no feasible combination of values for remaining variables can be found, but this cannot always be achieved with reasonable efficiency. The constraints (or, more precisely, their algorithms) interact through shared variables

(or, more precisely, their domains). Accordingly, our model for the problem at hand is composed of such constraints.

This filtering out of inconsistent values will not be sufficient in general to obtain a solution directly. Search is necessary, usually taking the form of an enumeration tree whose branches correspond to fixing a variable to a value in its current domain. Many strategies for choosing the next variable to fix and the next value to try have been devised and are often problem dependent. At each node of the tree, more filtering can occur since at least one of the domains has shrunk. For more details about constraint programming see [12].

The following notation is used throughout the paper.

Sets :

- I, the set of BTSs;
  - α_i, the capacity (in circuits) of the BTS $i \epsilon I$;
  - η_i the number of links necessary to connect it to a BSC;
- J, the set of BSC sites;
- K, the set of MSC sites;
- L, the set of links types;
  - β_ℓ, the capacity (in circuits) of a link of type $\ell \epsilon L$;
- S, the set of BSC types;
  - α_s, the capacity (in circuits) of a BSC of type $s \epsilon S$;
  - η_s^I, the maximum number of BTS interfaces of a BSC of type $s \epsilon S$;
  - η_s^T, the maximum number of MSC interfaces of a BSC of type $s \epsilon S$;
- T, the set of MSCs types;
  - α_t, the capacity (in circuits) of a MSC of type $t \epsilon T$;

– η_t^S the maximum number of BSC interfaces of a MSC of type $t \epsilon T$.

Note that all of these sets are disjoint.

Bandwidth parameters :

- $g_{i\,i'}$, the average number of communications per hour from BTS $i \epsilon I$ to BTS $i' \epsilon I$;
- $g_{i\,P}$, the average number of communications per hour from BTS $i \epsilon I$ to the public network;
- $g_{P\,i}$, the average number of communications per hour from the public network to BTS $i \epsilon I$.

Cost parameters :

- $a_{i\,j}$, the link and interface card costs (including the installation cost) for connecting BTS $i \epsilon I$ to a BSC at site $j \epsilon J$;
- $b_{\ell\,j}$, the link and interface card costs (including the installation cost) for connecting a BSC at site $j \epsilon J$ to an MSC at site $k \epsilon K$ through a link and interfaces of type $\ell \epsilon L$;
- c_j^s the cost of a BSC of type $s \epsilon S$ and installing it at site $j \epsilon J$;
- d_k^t, the cost of an MSC of type $t \epsilon T$ and installing it at site $k \epsilon K$.

Decision variables :

- v_i, the BSC site assigned to the BTS $i \epsilon I$;
- w_j, the MSC site assigned to site $j \epsilon J$;
- x_j, the BSC type installed at site $j \epsilon J$;
- y_k, the MSC type installed at site $k \epsilon K$;
- $z_{j\,l}$, the number of links of type $\ell \epsilon L$ installed from the BSC site $j \epsilon J$ to a MSC.

Traffic variables :

- t_i, the traffic (in erlang) from BTS $i \epsilon I$ to a BSC;
- t_j, the traffic (in erlang) from BSC $j \epsilon J$ to a MSC.

2.1 The Model

Constraints (C1) and (C2) are already enforced by the choices of decision variables. For example, the v_i variables ensure that the BTS i is connected to exactly one BSC. Indeed, the variables can only be assigned to one value of their domain. The others constraints are explicitly enforced in the model.

The constraint programming model for the design problem of wireless cellular networks, noted DPWCN, can now be given:

$$\min \left(\sum_{i \epsilon I} a_{i\, v_i} + \sum_{j \epsilon J} \sum_{\ell \epsilon L} z_{j\, \ell} b_{\ell\, w_j} + \sum_{j \epsilon J} c_j^{x_j} + \sum_{k \epsilon K} d_k^{y_k} \right) \tag{1}$$

where $a_{i\, v_i}$ is the link and interface card costs for connecting BTS $i \epsilon I$ to a BSC at site $v_i \epsilon J$; $b_{\ell\, w_j}$ is the link and interface card costs for connecting a BSC at site $j \epsilon J$ to an MSC at site $w_j \epsilon K$ through a link and interfaces of type $\ell \epsilon L$; $c_j^{x_j}$ is the cost of a BSC of type $x_j \epsilon S$ and of installing it at site $j \epsilon J$ and $d_k^{y_k}$ the cost of an MSC of type $y_k \epsilon T$ and of installing it at site $k \epsilon K$.

subject to

BSC capacity constraints (BTS interface level)

$$\sum_{i \epsilon I} ((v_i = j) \eta_i) \leq \eta_{x_j}^I \qquad (j \epsilon J) \tag{2}$$

where $(v_i = j)$ is 0-1 expression such that $(v_i = j)$ is equal to 1 if and only if $v_i = j$.

BSC capacity constraints (MSC interface level)

$$\sum_{\ell \epsilon L} z_{j\, \ell} \leq \eta_{x_j}^T \qquad (j \epsilon J) \tag{3}$$

BSC capacity constraints (switch fabric level)

$$\sum_{i \epsilon I} ((v_i = j) \alpha_i) \leq \alpha_{x_j} \qquad (j \epsilon J) \tag{4}$$

MSC capacity constraints (BSC interface level)

$$\sum_{j\epsilon J}((w_j = k)\sum_{\ell\epsilon L} z_{j\,\ell}) \leq \eta_t^S y_k \qquad (k\epsilon K) \tag{5}$$

MSC capacity constraints (switch fabric level)

$$\sum_{j\epsilon J}((w_j = k)\sum_{\ell\epsilon L} z_{j\,\ell}\beta_\ell) \leq \alpha_{y_k} \qquad (k\epsilon K) \tag{6}$$

Link capacity constraints

$$t_i \leq \alpha_i \qquad (i\epsilon I) \tag{7}$$

$$t_j \leq \sum_{\ell\epsilon L} z_{j\,\ell}\beta_\ell \qquad (j\epsilon J) \tag{8}$$

Traffic flow conservation constraints

$$t_j = \sum_{i\epsilon I}((v_i = j)\sum_{o\epsilon I}((v_o \neq j)(g_{i\,o} + g_{o\,i})) + \sum_{i\epsilon I}((v_i = j)(g_{i\,P} + g_{P\,i})) \qquad (j\epsilon J) \tag{9}$$

$$t_i = \sum_{o\epsilon I}(g_{i\,o} + g_{i\,P}) + \sum_{o\epsilon I} g_{o\,i} + g_{P\,i} \qquad (i\epsilon I) \tag{10}$$

Domains of the variables

$$v_i\epsilon J(i\epsilon I), w_j\epsilon K\ (j\epsilon J), x_j\epsilon S\ (j\epsilon J), y_k\epsilon T\ (k\epsilon K), Z_{j\,\ell}\epsilon N\ (j\epsilon J, \ell\epsilon L) \tag{11}$$

The cost function (1) of DPWCN, representing the total cost of the network sub-system, is composed of the cost of the links and interface cards and the cost of the BSCs and MSCs (including the installation costs). Constraints (2) require that the total number of BTS to BSC links connected at site $j\epsilon J$ be less than or equal to the maximum number of BTS interfaces that can be installed in the BSC type set up at that site. Constraints (3) dictate the total number of BSC to MSC links connected at site $j\epsilon J$ be less than or equal to the maximum number of MSC interfaces that can be installed in the BSC type set up at that site and constraints (4) impose the sum of the BTS rates connected to the BSC type installed at site $j\epsilon J$ be less than or equal to its switch fabric capacity. Constraints (5) dictate the total number of BSC to MSC links connected to site $k\epsilon K$ be less than or equal to the maximum number of BSC interfaces that can be installed in the MSC type set up at

that site and constraints (6) impose the sum of the rates of the BSC to MSC links connected to the MSC type installed at site $k \epsilon K$ be less than or equal to its switch fabric capacity. Constraints (7) and (8) are respectively BTS to BSC and BSC to MSC link capacity constraints and constraints (9) and (10) are the traffic flow conservation constraints.

3 The Heuristic Method

In this section, we present a local search heuristic for DPWCN, denoted CPLS (Constraint Programming Local Search). First, we decompose the model into two subproblems. The purpose of the first subproblem is to assign the BTSs to the BSCs and to select the BSC types at each BSC site for a given subset of BSC sites to use. The purpose of the second subproblem is to assign the BSCs to the MSCs, to select the MSC types and the BSC to MSC link types and to ensure traffic flot conservation. For a given subset the BSC sites, the subproblems are solved one after the other using a constraint programming software (e.g., ILOG OPL Studio [9]). The purpose of this decomposition is to avoid propagation (reduction of domain) on the traffic flow conservation variables during the assignment of $v_i's$. Such a propagation would be too time consuming because of a double summation in equation (9) that involves $v_i's$. Indeed, at each assignment of a v_i variable, equation (9) will be used to reduce the domain of t_j. To accomplish this task we should find the BTSs that are assigned to the same BSC site by calculating the double summation.

3.1 The Search Strategy

In constraint programming, the order in which the variables and the values are assigned is the search heuristic. The variables' assignment order has an impact on the topology of the search tree, the propagation and the efficiency of the search procedure. In this section, we will present the search strategy for DPWCN. The main idea behind this strategy is to estimate quickly the cost of a subset of BSC sites.

In the first subproblem, corresponding to the section between BTS and BSC, the assignment heuristic first fixes each BTS to a BSC site (v_i) (Figure 2). In the figures the "forall" keyword is used to specify the order in which variables are assigned and the "tryall" keyword is used to specify

Strategy 1 : BTS to BSC site assignment
forall($i \epsilon I$ *ordered by decreasing* η_i)
tryall($j \epsilon J$ *ordered by increasing* d_{ij})
$v_i = j$

Figure 2: The strategy of the first subproblem

the order in which the values are tried. The BTS variables are ordered by decreasing number of links used, to assign first the BTS with more links and reduce the total link cost. The values in the domain of each BTS variable v_i are ordered by increasing distance with each BSC site. Hence, the BTSs are first assigned to the nearest BSC site available. Next, we can determine the BSC type to install at each site. Knowing the BTS assignment it is easy to install the least cost BSC type at each site (x_j). For each BSC site, we compute the minimum cost equipment to install by adding the demands of the BTSs assigned to the BSC site.

The second subproblem, the network between the BSCs and the MSCs, has small instances, allowing us to evaluate each possibility. The solutions found for this section are optimal according to the choices made in the first subproblem. The search procedure first determines the MSC type to install at each site (y_k) considering the demand of each BSC (Strategy 2, Figure 3). The variables are assigned in simple lexicographic order because we evaluate all the possibilities so we don't need to begin with a particular variable. As mentioned before, by ordering the values by increasing cost, the first solution obtained for this variable is the least expensive one. Finally, we determine the number of links between the BSC and MSC sites (z_{jl}) (Strategy 3, Figure 3). The variables are ordered by increasing link cost and the values are ordered by increasing number of links to install, consequently the first feasible solution found is the least expensive one. We use a time limit (TL) because it is possible that there is no solution. This time limit was determined experimentally (keyword "timeLimit" in the strategy). The z_{jl} assignment will also fix the BSC assignment (w_j) : the BSC site will be assigned to the MSC site where there are links between them.

Strategy 2 : MSC type to MSC site assignment $forall(k \epsilon K)$ $tryall(t \epsilon T$ *ordered by increasing* $cost_t)$ $y_k = t$
Strategy 3 : Number of links of each type between the BSC and the MSC assignment $timeLimit(TL)$ $forall(j \epsilon J)$ $forall(\ell \epsilon L$ *ordered by increasing* $cost_\ell)$ $tryall(n \epsilon 0..max_\ell)$ $z_{j\ell} = n$

Figure 3: The strategies of the second subproblem

3.2 Probing

The PC model estimates the cost of a open subset of BSC sites. Now, we should find the subset of BSC sites that leads to the least expensive solution. We use a local search with a simple descent algorithm [7] to do it. The search space of the local search is limited to the BSC sites used in the solution. The goal is to determine the BSC sites that will be part of the final solution. The PC model is used by the local search to evaluate the cost of the subsets of BSC sites. We start with an initial subset of BSC sites and evaluate it by solving each subproblem approximately (by fixing the computation time of the search). This is similar to launching a probe to approximate the quality of the solution.

3.3 The Local Search

The main idea behind the heuristic CPLS is to find a good subset of BSC sites. Each BTS is assigned to the nearest BSC site available, so the BSC sites with many BTS assigned to them are better for the optimisation of the cost of BTS to BSC links. As a matter of fact, we keep those BSC sites in the solution during the local search procedure, and thus try to remove those having the lowest number of BTS assigned to them. The local search procedure (Algorithm 1) begins by ordering the BSC sites in the current solution according to the number of BTS assigned to each one. After that, at each iteration of the local search, we select the number of BSC sites, denoted n, used in the current solution having the lowest number of BTS assigned to them. To

visit the neighborhood $N(x_k)$ of the current solution x_k at the iteration k, we evaluate the cost f_x of solutions obtained by changing one of the n sites previously identified ($i \epsilon n$) by another not used in the current solution as well as the solution obtained by removing the that site. The best solution is selected as the current solution. The local search continues while the cost of the current solution decreases.

Algorithm 1 Local Search Algorithm

```
o ← number of BSC sites used by the model
J ← set of possible BSC sites
n ← number of BSC sites considered for removal
m ← number of BSC sites considered for addition
repeat
   sort J in increasing order of the number of BTS assigned
   J' ← the first o sites from J
   J'' ← J\J'
   for i in 1...n do
      remove the i^th site of J'
      launch the probe to estimate the neighbour's cost
      for j in 1...m do
         replace the i^th site of J' by the j^th of J''
         launch the probe to estimate the neighbour's cost
      end for
   end for
   compare the probes and keep the least expensive solution
until ∄x ϵ N(x_k) such that f(x) ≤ f(x_k)
```

Algorithm 2 Initial Solution Algorithm

```
o ← number of BSC sites used by the model
p ← number of BTSs
q ← number of sites to permute
sort J in increasing order of sum of distances with p BTS
J' ← the first o sites from J
launch the probe on J' to determine the cost
repeat
   permute q sites of J' with q sites not in J'
   launch the probe on J' to determine the cost
until ∄ some possible distinct permutations
return J' of better cost
```

Table 1: Features of the BTS types.

	Type A	Type B	Type C
Capacity (circuits)	96	288	576
Number of BTS DS-1 interfaces	1	3	6

Table 2: Costs of the BSC types (including the installation costs.

	Type A	Type B	Type C
Capacity (circuits)	5000	10 000	15 000
Maximum number of BTS interfaces	15	30	60
Maximum number of MSC interfaces	15	30	60
Cost ($)	50 000	90 000	120 000

3.4 The Initial Solution

The heuristic CPLS begins by looking for an initial solution. The initial subset is selected by considering the closest BSC site to the BTSs. We order the BSC sites according to the sum of distances with the p nearest BTS (Algorithm 2). We choose the o BSC sites for which p BTSs will be assigned with the lower cost. Then, we change q BSC sites of this subset by q others to evaluate the impact of each BSC site. We keep the best subset as the initial solution. The parameters used to search for the initial solution were obtained by experimentation.

4 Numerical Results

In this section, we evaluate the performance of the proposed heuristic. The heuristic was implemented in the ILOG OPL Script language [9] on a AMD Athlon PC (1.6 GHz and 512 MB of RAM). For the tests, three BTS types, three BSC types and three MSC types are used. Their features are presented respectively in Tables 1, 2 and 3. Moreover, DS-1 links are used to connect the BTSs to the BSCs and DS-1 and DS-3 links are used to connect the BSCs to the MSCs. The interface costs are presented in Table 4 and the link costs in Tables 5 and 6. Table 7 presents the parameters used to find the initial solution.

For the tests, 28 instances were generated as follows. $|I|$ points corresponding to BTSs' locations,

Table 3: Costs of the MSC types (including the installation costs).

	Type A	Type B	Type C
Capacity (circuits)	100 000	200 000	300 000
Maximum number of BSC interfaces	50	100	150
Cost ($)	200 000	350 000	500 000

Table 4: Costs of the interface types (including the installation costs).

Interface type	Capacity (circuits)	Cost ($)
DS-1	96	500
DS-3	2688	2500

Table 5: Costs of the BTS to BSC links (including the installation costs).

BSC type	Number of DS-1	Capacity (circuits)	Cost($)
A	1	96	2000
B	3	288	3000
C	6	596	4000

Table 6: Costs of the BSC to MSC links (including the installation costs).

Link type	Capacity (circuits)	Cost ($/km)
DS-1	96	2000
DS-3	2688	4000

Table 7: Initial heuristic parameters.

$\|I\|$	o	p	q
50	4	12	2
100	4	18	2
150	4	24	3
200	4	28	3

$|J|$ points corresponding to the BSC sites and $|K|$ points corresponding to the MSC sites were generated in a 100 km by 100 km grid following a uniform distribution. The type of each BTS was selected randomly among the three BTS types considered (Table 1). Finally, the demand between each pair of BTSs and between the BTSs and the public network was generated randomly in the interval [0,0.2] erlang, following a uniform distribution. All test results were compared to the results obtained on the same instances with a tabu search algorithm, denoted TS, proposed by Chamberland and Pierre [5]. The results are presented in Table 8. In this table, Columns 1 to 3 present respectively the number of BTSs, the number of BSC sites and the number of MSC sites. Column 4 presents a lower bound, denoted LB, obtained by solving a relaxed version of the model which does't consider the traffic flow conservation constraints (see [5] for more detail concerning the lower bound). Columns 5 and 6 present respectively the results obtained by the TS algorithm and the corresponding CPU time. Columns 6 to 10 present the value of the solution found by algorithm CPLS, the CPU time to find it and the GAP that indicates the percentage of improvement of the solution value compared to the value of LB and TS. These results show that CPLS confirm and improve, on average, the solutions found by TS. Indeed, the average improvement is 0.17% and TS has found only eight better solutions than CPLS. Moreover, the average percentage gap between the solution values found by CPLS and the lower bound values is 5.13%.

As we can see in Table 8, the CPU time used by the CPLS approach is higher than the one used by TS algorithm. In this context of a problem at the strategic level, the quality of the solution found is more important than the speed of the algorithm. In any case, figure 5 illustrates that the initial solutions quickly compare to the final solutions.

5 Conclusion

In this paper, we studied the tree-topology design problem for wireless cellular networks that consists of selecting the location of the network nodes (BSC and MSC) and their types, designing the network tree-topology and selecting the link types. A constraint programming model has been proposed combined with a local search heuristic.

As mentioned before, the purpose of the heuristic strategies is to estimate quickly the cost of

Table 8: Computational results

$\|I\|$	$\|J\|$	$\|K\|$	LB	TS		LS			
			OBJ	OBJ	CPU	OBJ	CPU	GAP (%)	
			(k$)	(k$)	(sec)	(k$)	(sec)	vs LB	vs TS
50	10	10	4051.5	4083.5	91	4051.5	360	0.00	-0.78
		20	4148.0	4218.0	159	4177.7	681	0.72	-0.96
	20	10	3316.6	3321.1	126	3321.1	2033	0.14	0.00
		20	3557.4	3583.2	182	3559.1	2445	0.05	-0.67
	30	10	3518.3	3525.2	176	3518.3	6147	0.00	-0.20
		20	3583.8	3591.7	245	3591.7	4092	0.22	0.00
	40	10	3267.9	3304.2	254	3284.5	9968	0.51	-0.60
		20	3318.8	3366.1	363	3337.6	12076	0.57	-0.85
100	10	10	6865.3	7215.8	200	7136.3	718	3.95	-1.10
		20	6573.1	6944.9	320	6861.0	517	4.38	-1.21
	20	10	6244.2	6599.4	220	6576.9	4485	5.33	-0.34
		20	6934.9	7150.9	417	7112.8	4730	2.57	-0.53
	30	10	6114.5	6217.3	381	6157.9	14049	0.71	-0.96
		20	5606.1	5804.1	614	5737.9	11756	2.35	-1.14
	40	10	5399.7	5607.3	573	5586.6	16400	3.46	-0.37
		20	5249.7	5377.7	731	5364.8	18807	2.20	-0.24
150	20	10	8571.8	9346.3	370	9330.1	5374	8.85	-0.17
		20	8208.2	8913.1	774	8924.9	6210	8.73	0.13
	30	10	7725.0	8294.5	593	8289.4	13903	7.31	-0.06
		20	7551.4	8092.9	759	8109.3	17120	7.39	0.20
	40	10	8234.0	8817.5	885	8739.4	60179	4.86	-0.89
		20	7360.4	8037.9	1086	8064.9	41513	9.57	0.34
200	20	10	10608.2	11843.7	492	11811.4	13827	11.34	-0.27
		20	11106.3	12231.6	862	12295.7	7533	10.71	0.52
	30	10	9281.2	10216.8	709	10384.6	40764	11.89	1.64
		20	10304.1	11070.6	1030	11312.0	25234	9.78	2.18
	40	10	9111.7	10030.0	1158	10123.4	87318	11.10	0.93
		20	9598.3	10467.4	1489	10549.0	55555	9.90	0.78

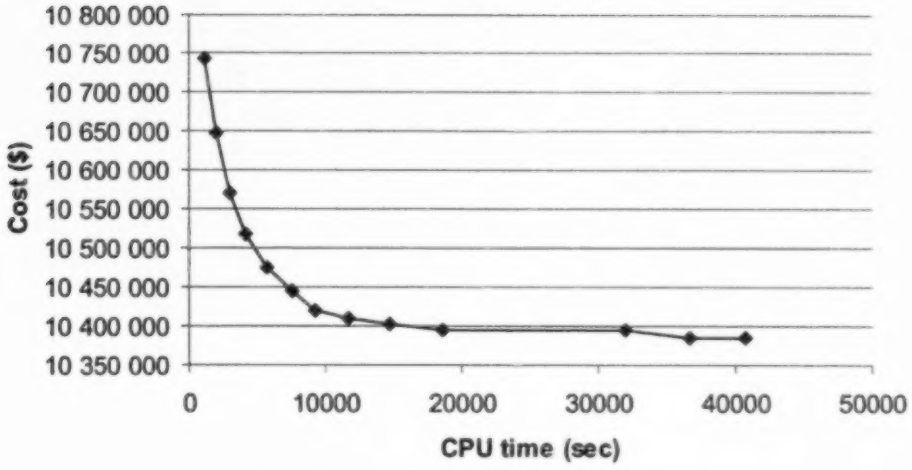

Figure 4: Typical evolution of the cost function (200 BTS, 30 BSC, 10 MSC)

a subset of BSC sites. In Strategy 1, it is possible that some reassignment reduces the cost of the solution. This is explained by the fact that some BTSs could not be assigned to the nearest BSC site because of the capacity limit of a BSC. This possible reassignment provides the biggest uncertainty of our estimation method. However, in a real network, the BTS and the BSC sites are geographically distributed. So, it is possible that some BSC site has a demand exceeding their capacity but not in an excessive way. The BTSs that were not assigned to the nearest BSC site will be assigned to the second nearest one. The possible gain for a real network is small. The other strategies find the optimal solution according to the choices made by the first strategy. So the estimation is good for a real network.

Experimental results show that the local search heuristic produces, on average, better solutions than the best previously known algorithm for this problem. Since our local search approach is essentially a simple descent and therefore runs the risk of getting trapped in a local minimum, improvements could possibly be achieved through standard mechanisms to escape such minima [7, 8, 10]. A first attempt with a multi-start approach (multiple runs of the algorithm with a

different initial solution) improved fewer than 10% of the solutions and by only 0.5% on average, at the expense of a higher computational cost. Similarly, preliminary experiements with variable neighbourhood search only improved some of the solutions and by less than 1%.

References

[1] Ahuja, R., Magnanti, T., and Orlin, J.,"Network Flows", Prentice Hall, 1992.

[2] Andr, M., Pesant, G. and Pierre, S., "A Variable Neighborhood Search Algorithm For Assigning Cells to Switches in Wireless Networks", submitted, 2002.

[3] Amoussou, G., Andr, M., Pesant, G. and Pierre, S., "A constraint programming approach to assign cells to switches in personnal communication networks", Annals of Telecommunications, vol. 58, no. 3, pp. 584-604, 2003.

[4] Bhattacharya, P.S., Saha, D. and Mukherjee, A., "Heuristics for assignment of cells to switches in a PCSN: a comparative study", IEEE International Conference on Personal Wireless Communications, Jaipur, India, pp. 331-334, 1999.

[5] Chamberland, S. and Pierre, S., "On the Design Problem of Cellular Wireless Networks", ACM/Kluwer Wireless Networks, to appear.

[6] Cox, L.A. Jr. and Sanchez, J.R., "Designing least-cost survivable wireless backhaul networks", Journal of Heuristics, vol. 6, no. 4, pp. 525-540, 2000.

[7] Glover, F. and Laguna, M., "Tabu Search", Kluwer Academic Publisher, 1997.

[8] Hansen, P., Mladenovic, N. and Perez-Brito, D., "Variable Neighborhood Decomposition Search", Journal of Heuristics, vol. 7, no. 4, pp. 335-350, 2001.

[9] ILOG, "ILOG OPL STUDIO 3.5 - The Optimisation Language", 1999.

[10] Kirpatrick, S., Gelatt, C.D. Jr. and Vecchi, M.P.,"Optimization by Simulated Annealing", Science, vol. 220, pp. 671-680, 1983.

[11] Jonker, R. et Volgenant T., "A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems", Computing, vol. 38, pp. 325-340, 1987.

[12] Marriott, K. and Stuckey, P.J., "Programming with Constraints : An Introduction", The MIT Press, 1998.

[13] Merchant, A., and Sengupta, B., "Multiway graph partitioning with applications to PCS networks", IEEE INFOCOM'94, vol. 2, pp. 593-600, 1994.

[14] Merchant, A., and Sengupta, B., "Assignment of cells to switches in PCS networks", IEEE/ACM Transactions on Networking, vol. 3, no. 5, pp. 521-526, 1995.

[15] Pierre, S. and Houto, F., "Assigning Cells to Switches in Cellular Mobile Networks Using Taboo Search", IEEE/ACM Transactions on System, Man and Cybernetics, vol. 32, no. 3, pp. 465-478, 2002.

[16] Pierre, S. and Quintero, A., "Affectation de cellules aux commutateurs dans les rseaux mobiles: interactions des algorithmes gntiques, de recuit simul et de recherche taboue", submitted, 2001.

[17] Saha, D., Mukherjee, A. and Bhattacharya, P., "A Simple Heuristic for Assigment of Cell to Switches in a PCS Network", Wireless Personal Communication, vol. 12, pp. 209-224, 2000.

[18] Sohn, J. and Park S., "Efficient solution procedure and reduced size formulations for p-hub location problems", European Journal of Operational Research, vol. 108, pp. 118- 126, 1998.